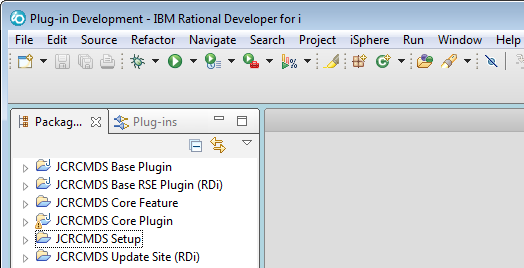
Initial Upload of JCRCMDS to SourceForge

# Prerequisites

Installed IBM Rational Developer for i (RDi 9.5) with a dedicated workspace for JCRCMDS and the JCRCMDS plug-ins (Selected perspective: Plug-in Development):



This tutorial uses the following workspace location:

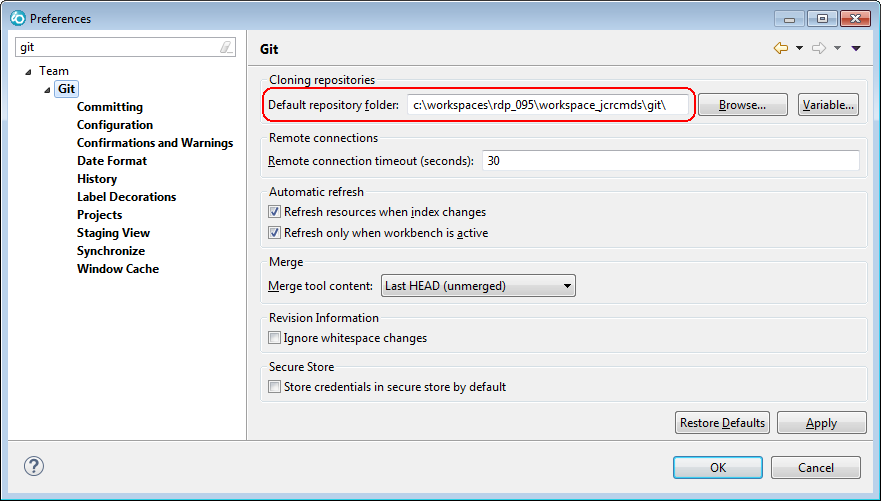
c:\workspaces\rdp\_095\workspace\_jcrcmds\

# Setting Up Git

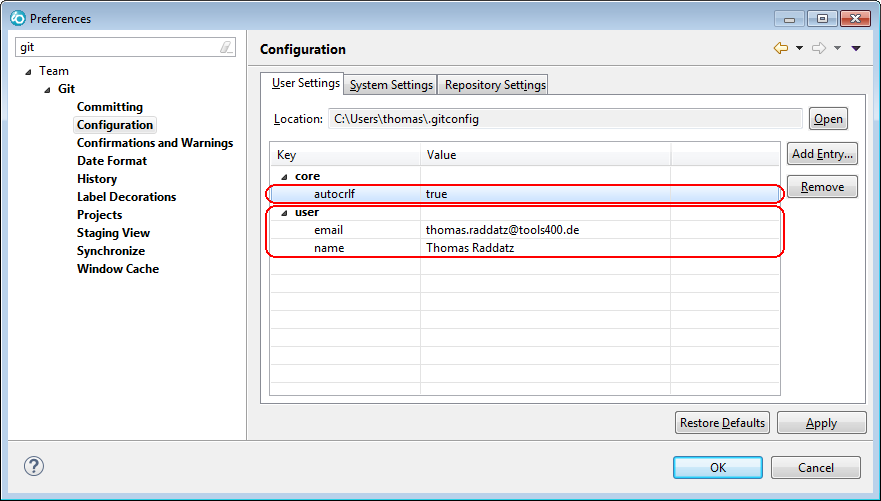
Open the Git preferences page and set the following values:

**Default Repository Folder**

c:\workspaces\rdp\_095\workspace\_jcrcmds\git\



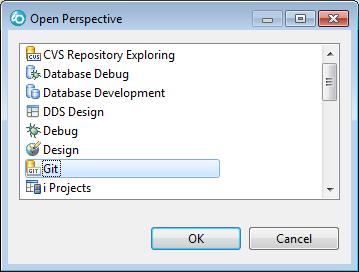
**CRLF Handling And User**



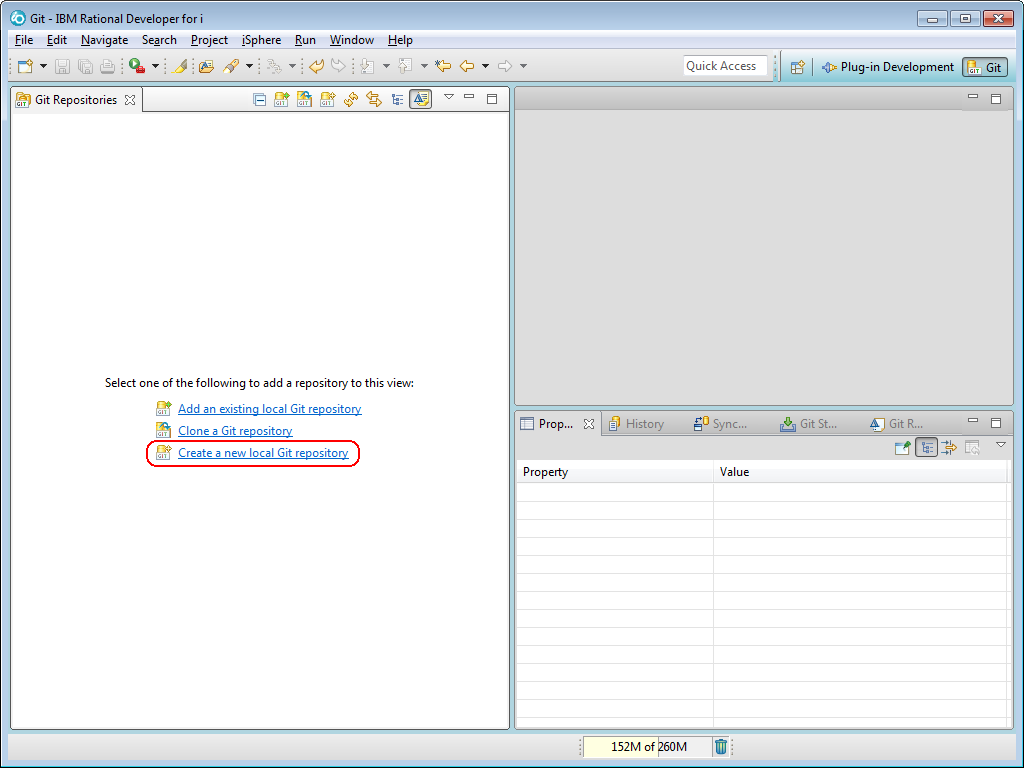
Of course you must not use my email address and name, but your own address and name.

# Creating a Local Git repository

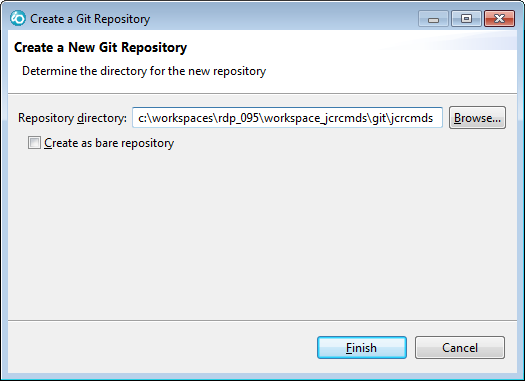
Select the "Git" perspective to create a local Git repository for JCRCMDS:



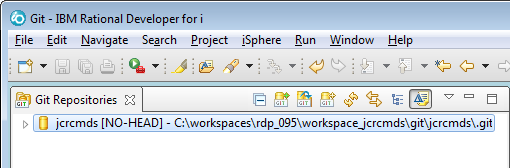
On the Git perspective select "Create a new local Git repository":



Enter the path of the new directory. In this example we create a new folder "jcrcmds" below the root Git path:

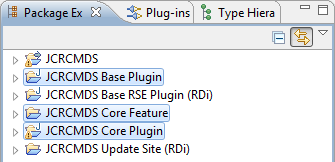


You should get the following repository entry in the Git perspective:

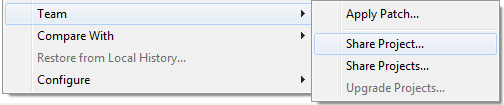


# Adding Projects to Git

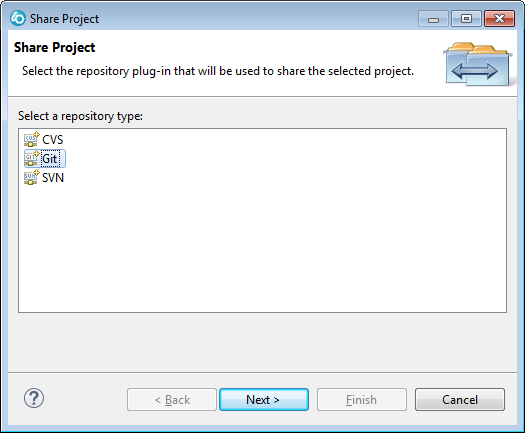
Select the JCRCMDS projects that go into the "eclipse" sub-path of your Git repository. We want to separate the plug-ins that are common for RDi and WDSCi from the plug-ins that are specific to the IBM IDEs:



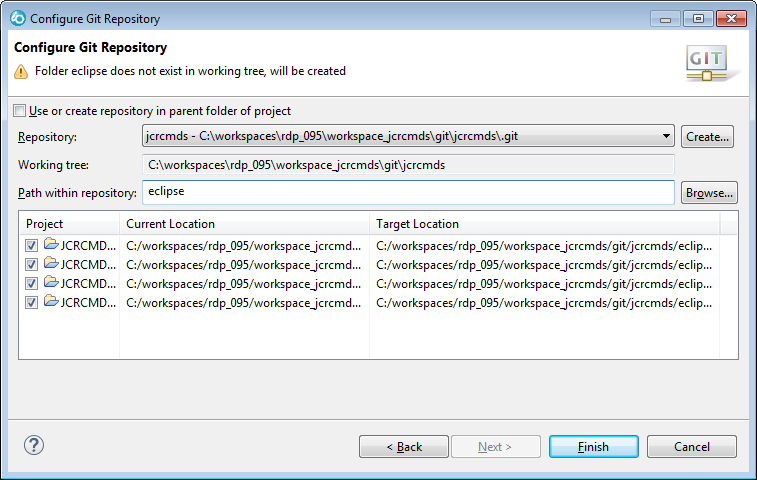
Open the context menu and select "Team - Share Project…":



Then continue with "Git":



Select the "jcrcmds" repository and set the "Path within repository" to "eclipse":

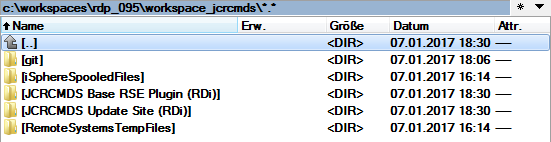


Click the [Finish] button.

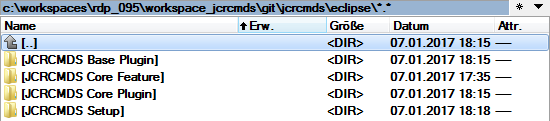
The question marks next to the projects indicate that the projects have been added to Git and that there are uncommitted changes.

The projects have been moved from the workspace to the git repository:

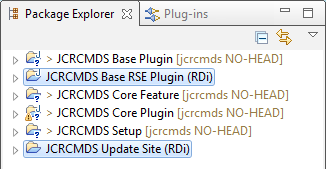
**Workspace**

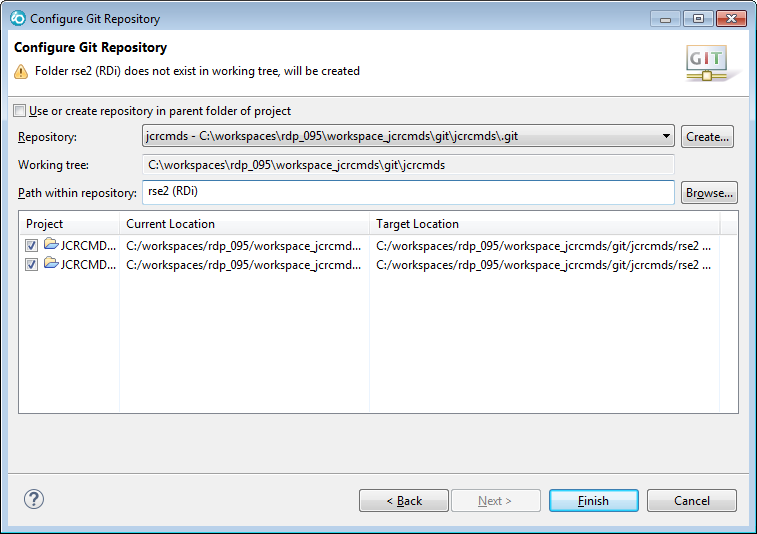
****

**Local Git Repository**



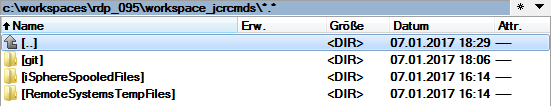
Go ahead and add the remaining IBM IDE specific projects to sub-folder "rse2 (RDi)". The WDSCi plug-in will go into folder "rse1 (WDSCi)":



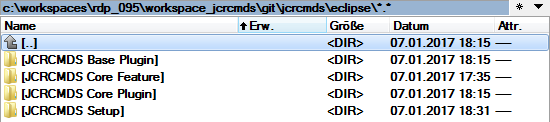


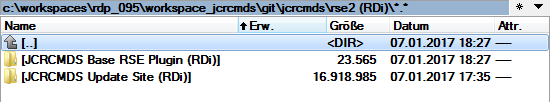
Now all projects are in the Git repository and the workspace is empty again:

**Workspace**



**Local Git Repository**





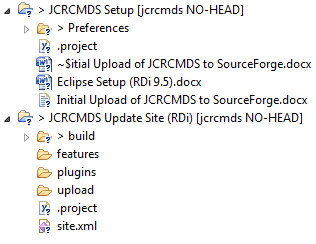
# Preparing The Initial Commit

Before we can commit our projects, we need to add a few items to the "ignore list", because we do not want these items go into the repository. These items are temporary work items:

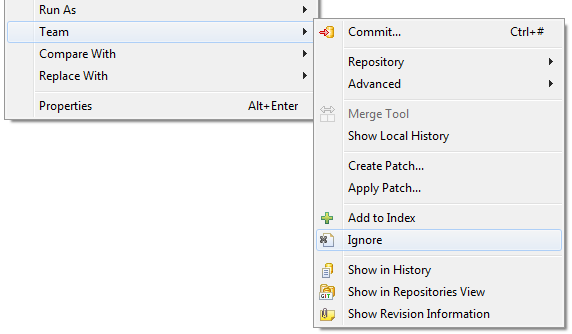
* MS Word temporary files
* Everything in the directories "features", "plugins" and "upload"

The MS Word file is there, because I write that tutorial while I practice it. Most likely you do not have temporary files when you are about for the first commit.

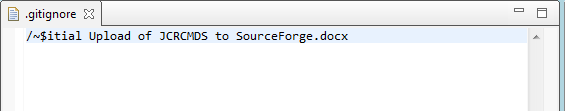
Let us start with the MS Word temporary files, which start with a tilde (~):



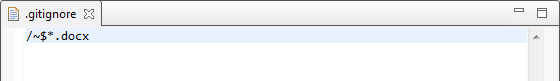
Select that file and add it to the "ignore list":



Now open the "Navigator" view and open the ".gitignore" file of project "JCRCMDS Setup":

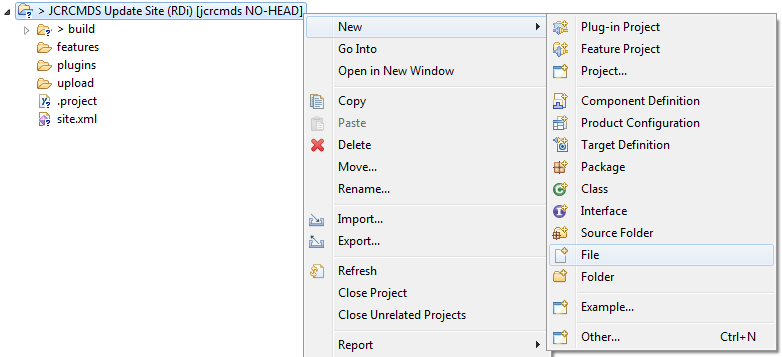


Change the item to a generic name like this:

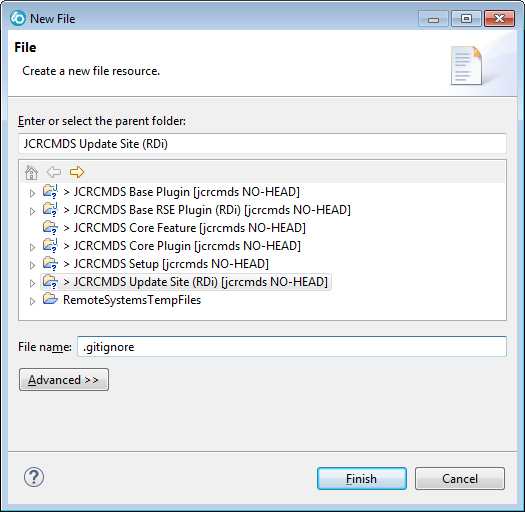


This way all MS Word temporary files are ignored.

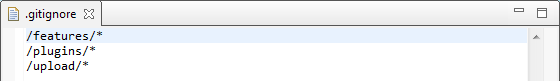
Now go ahead with the former mentioned directories of the "JCRCMDS Update Site" project. For this project a ".gitignore" file has not yet been created. Therefore you should create it yourself. Right click the project and select "New - File":



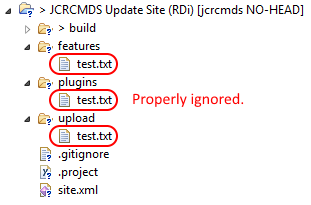
Enter ".gitignore" and click the [OK] button:



Open the file and add the following entries:



Now feel free to add some test files to these folders to see, that the files are ignored whereas the folders are not:

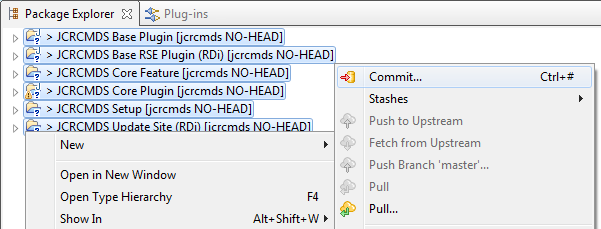


Removed the test files. You not need them any longer.

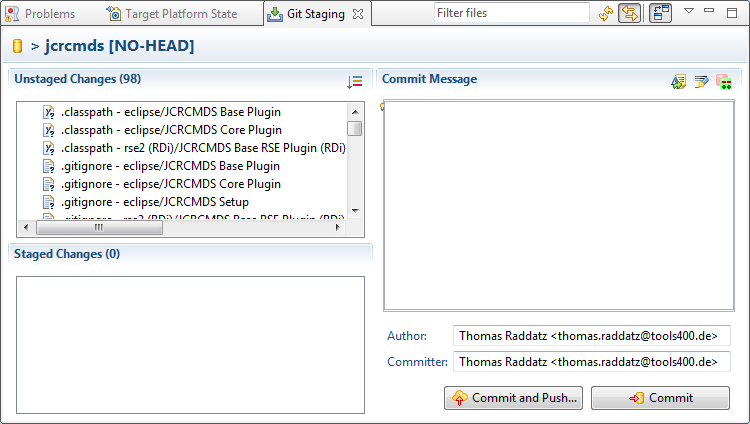
Now everything is set up for the first commit.

# Initial Commit

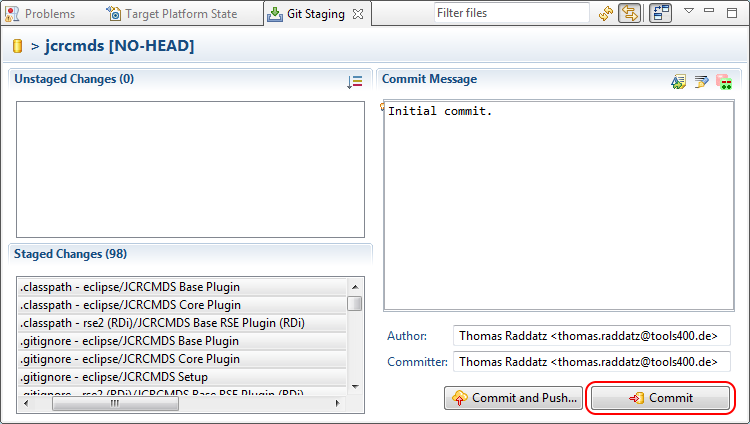
Select all projects, then commit your changes:



Notice, that the "Git staging" view has been opened. You may adjust its size to see all options:

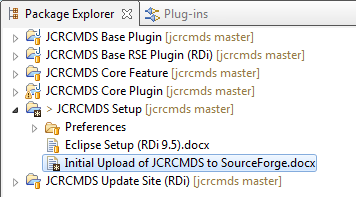


The "Unstaged Changes" window contains all uncommitted changes. In order to commit the changes you first need to move them to the "Staged Changes" window. Select one item and then press Ctrl+A to select all. Then drag and drop the items into the "Staged Changes" window or use the context menu and menu option "Add to index" to do so. Also add a commit message. The result should look like this:



Now click the [Commit] button to commit the changes to your local repository.

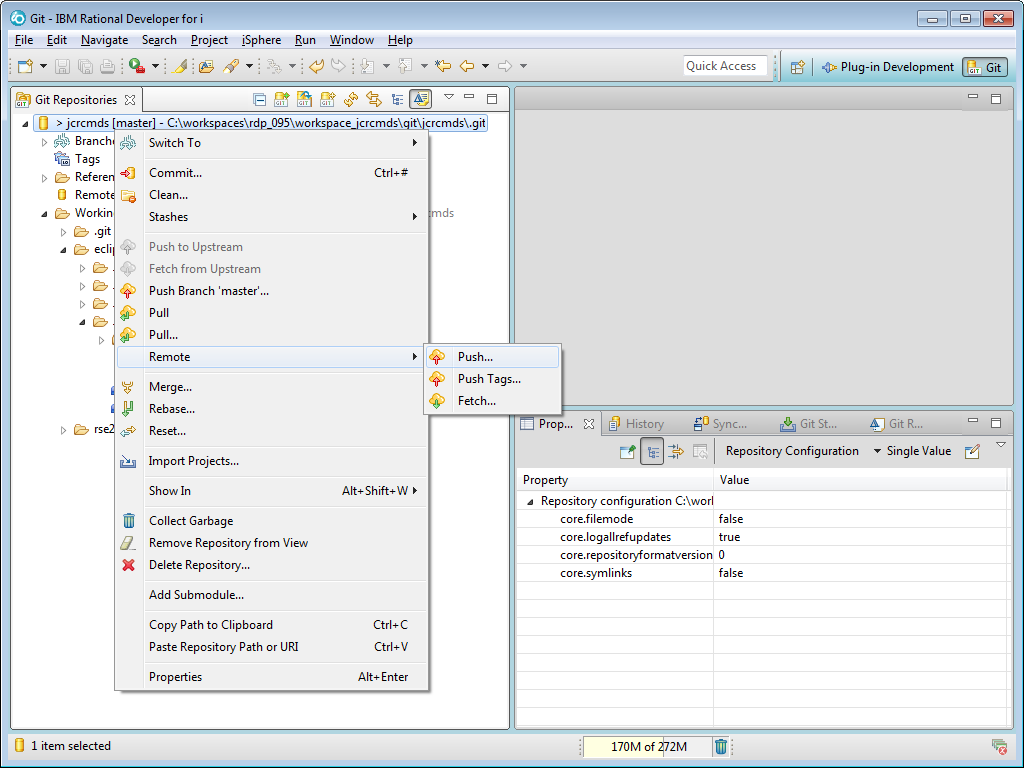
Now everything but "Initial Upload of JCRCMDS to SourceForge.docx" is marked as committed (the orange little decoration). The black asterisk decoration of "Initial Upload of JCRCMDS to SourceForge.docx" indiactes, that this files has uncommited changes again. That is, because I often update that file while writing this tutorial.



I also added "/~\*.tmp" to the Git ignore list of project "JCRCMDS Setup" to exclude these temporary files as well.

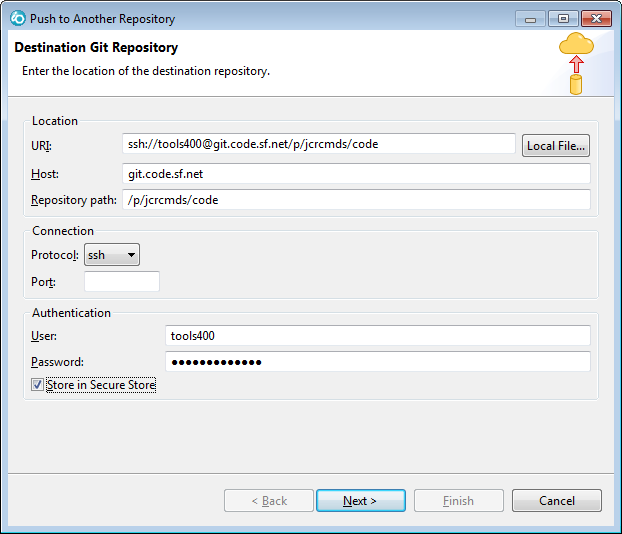
# Pushing Changes of Local Repository to SourceForge

Switch to the "Git" perspective to push the local Git repository to SourceForge for the first time.



Paste the URL of the SourceForge Git repository into the dialog that just popped up:

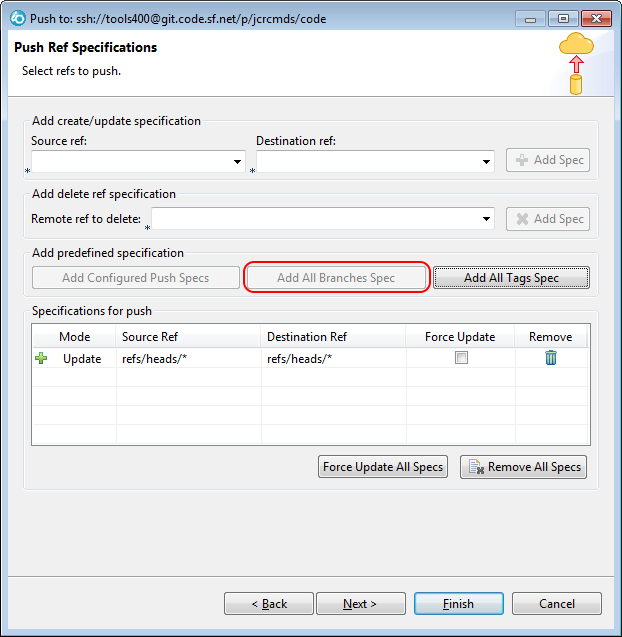
ssh://tools400@git.code.sf.net/p/jcrcmds/code



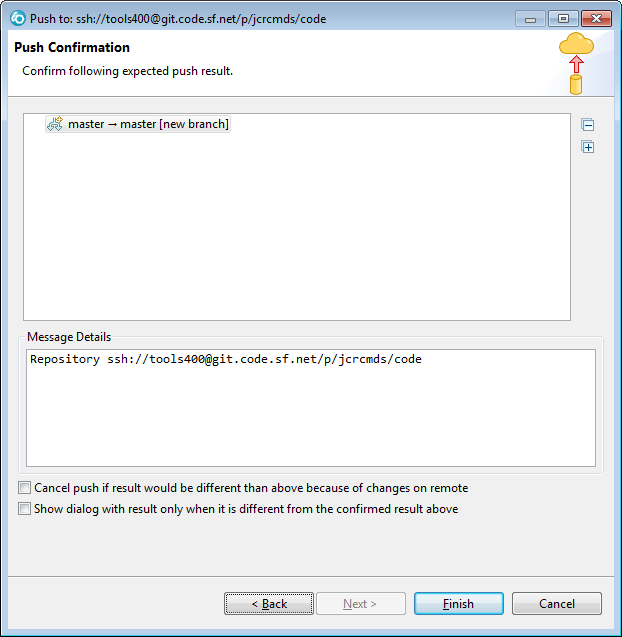
The user has been automatically retrieved from the global Git preferences.

Enter your SourceForge password and optionally check "Store in Secure Store". I usually do that, because I am a lazy developer and I do not like entering it again and again

On the next page click [Add All Branches Spec] and then [Next]:

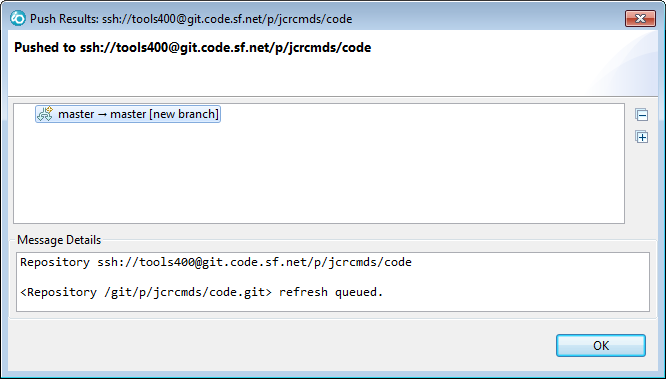


Now you are ready and about to push your projects to SourceForge.

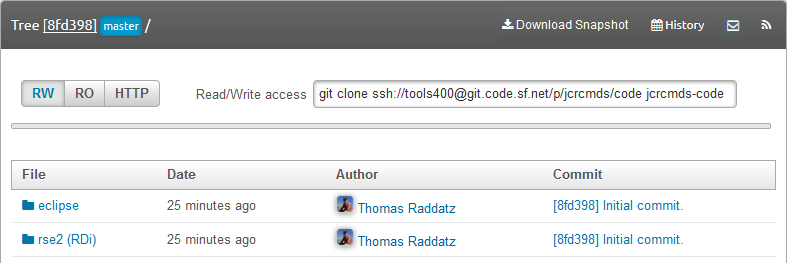


Click the [Finish] button to do so.

The following dialog confirms that it worked just fine:

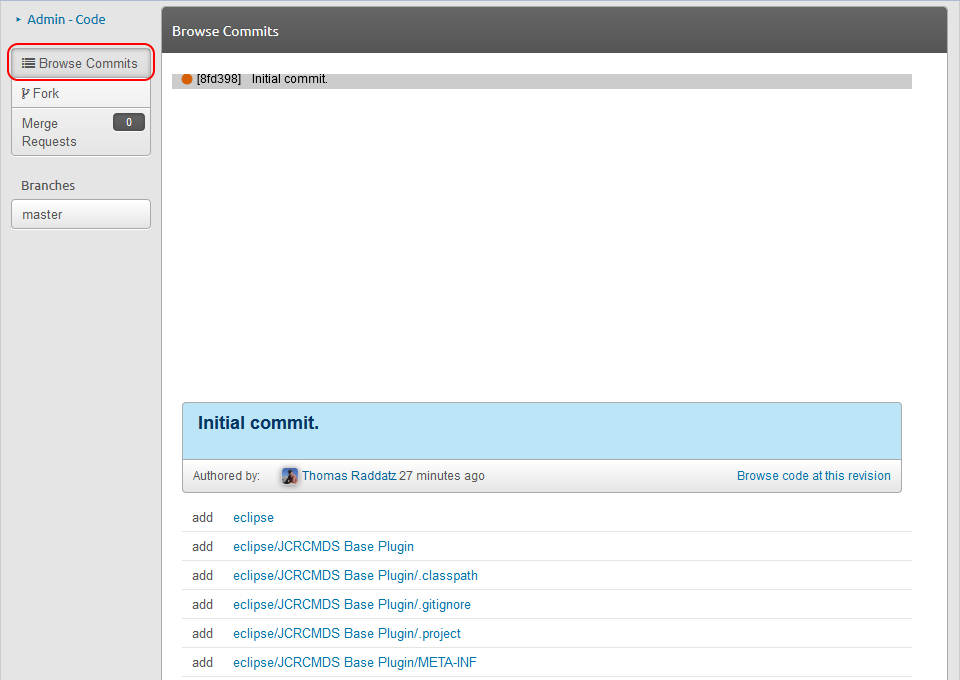


Let us check that at SourceForge:



Looks good, doesn't it?

Let us also check the commits:



Looks well, too.

Thomas Raddatz, 07.01.2017